COMP-767: Reinforcement Learning - Assignment 1

Posted Thursday, January 17, 2019
Due Tuesday, January 29, 2019

The assignment can be carried out individually or in teams of two. You have choices on both parts of the assignments.

1. **Bandit algorithms [50 points]**

   Choose **one** of the following topics.

   (a) Best-arm identification: also sometimes called *pure exploration*, this is a problem formulation in which the

   For this part of the assignment you will have to read the following paper:

   Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting, Kevin Jamieson and Robert Nowak, CISS, 2014:

   https://people.eecs.berkeley.edu/ kjamieson/resources/bestArmSurvey.pdf

   The paper describes two different classes of algorithms for this problem. Your task is to: (a) summarize the main results in the paper; (b) Reproduce the results in Figure 1 (c) Perform the same empirical comparison on the bandit problem provided in the Sutton & Barto book (which we discussed in class). Do not forget to average your results over multiple independent runs. (d) discuss in a short paragraph a concrete application in which you think regret optimization would be more useful than best arm identification.

   ① Summary of results of the paper.
   ② Reproduce fig 1
   ③ empirical comparison on the bandit problem
   L ^ ^ ?
   ④ application where regret optimization is better for best arm identification.

   (b) We discussed in class Thompson sampling, which is a very useful Bayesian algorithm for bandits. We mentioned briefly regret results. In this part of the assignment, you should read the following paper by Agarwal and Goyal (2012), which provides early prior-independent regret bounds for Thompson sampling:

   https://arxiv.org/pdf/1209.3353v1.pdf

   You need to write a short summary (max 3 pages in latex format of your choice) of the result and the main steps and ideas in the proof. Feel free to add some background if you think it would be necessary to understand their approach.

2. **Markov Decision Processes and dynamic programming [50 points]**

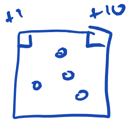
   Choose **one** of the following topics.

   (a) Implement and compare empirically the performance of value iteration, policy iteration and modified policy iteration. Modified policy iteration is a simple variant of policy iteration in which the evaluation step is only partial (that is, you will make a finite number of backups to the value function before an improvement step). You can consult the Puterman (1994) textbook for more information. You should implement your code in matrix form. Provide your code as well as a summary of the results, which shows, as a function of the number

   modified policy iter

empirically compare performance of value iteration, policy iteration & modified policy iteration.

Q2



of updates performed, the true value of the *greedy policy* computed by your algorithm, from the bottom left state and the bottom right state. That is, you should take the greedy policy currently considered by your algorithm and compute its exact value.

To test your algorithm, use a grid world in which, at each time step, your actions move in the desired direction w.p. p and in a random direction w.p. (1-p). The grid is empty, of size $n \times n$. There is a positive reward of +10 in the upper right corner and a positive reward of +1 in the upper left corner. All other rewards are 0.

You need to test your algorithm with two different values of p (0.9 and 0.7) and with two different sizes of gird ($n = 5$ and $n = 50$). Explain what you see in these results.

(b) We mentioned briefly in class the linear programming approach to solving MDPs. One reason why this approach is interesting is that it can work with either a primal or a dual formulation. For this part of the assignment, you will read a paper by Wang et al, which attempts to leverage these ideas but in dynamic programming:

https://era.library.ualberta.ca/items/6ef4eab6-acb1-48f3-a188-8c6f2aa39c2d/view/7145a384-5ea5-4d7e-abe0-d29e013b8921/TR07-10.pdf

You need to write a short summary (max 3 pages in latex format of your choice) of the intuition of their approach, what is their ”primal” and ”dual” formulation, and what are the main theoretical insights of the work. Feel free to add some background if you think it would be necessary to understand their approach. In one paragraph, please propose one improvement to this work.

background = duals in complex optimization.

Best-arm Identification Algorithms for Multi-Armed Bandits in the Fixed Confidence Setting

Kevin Jamieson and Robert Nowak
Department of Electrical and Computer Engineering
University of Wisconsin - Madison
Email: kgjamieson@wisc.edu and nowak@ece.wisc.edu

1a

***Abstract*—This paper is concerned with identifying the arm with the highest mean in a multi-armed bandit problem using as few independent samples from the arms as possible. While the so-called "best arm problem" dates back to the 1950s, only recently were two qualitatively different algorithms proposed that achieve the optimal sample complexity for the problem. This paper reviews these recent advances and shows that most best-arm algorithms can be described as variants of the two recent optimal algorithms. For each algorithm type we consider a specific instance to analyze both theoretically and empirically thereby exposing the core components of the theoretical analysis of these algorithms and intuition about how the algorithms work in practice. The derived sample complexity bounds are novel, and in certain cases improve upon previous bounds. In addition, we compare a variety of state-of-the-art algorithms empirically through simulations for the best-arm-problem.**

I. Introduction

fixed confidence setting

This paper describes recent advances in algorithms for identifying the arm with the highest mean in a stochastic multi-armed bandit (MAB) problem with high probability using as few total samples as possible. Consider a MAB with n arms, each with unknown mean payoff $\mu_1, \dots, \mu_n$ in $[0, 1]$. A sample of the ith arm is an independent realization of a sub-Gaussian random variable with mean μ_i. In the *fixed confidence setting*, the goal of the best arm problem is to devise a sampling procedure with a single input δ that, regardless of the values of $\mu_1, \dots, \mu_n$, finds the arm with the largest mean with probability at least $1 - \delta$. More precisely, best arm procedures must satisfy $\sup_{\mu_1,\dots,\mu_n} \mathbb{P}(\widehat{i} \neq i_*) \leq \delta$, where i_* is the best arm, $\widehat{i}$ an estimate of the best arm, and the supremum is taken over all set of means such that there exists a unique best arm. In this sense, best arm procedures must automatically adjust sampling to ensure success when the mean of the best and second best arms are arbitrarily close. Contrast this with the *fixed budget setting* where the total number of samples remains a constant and the confidence in which the best arm is identified within the given budget varies with the setting of the means. While the fixed budget and fixed confidence settings are related (see [1] for a discussion) this paper focuses on the fixed confidence setting only.

alt setting

The best arm problem has a long history dating back to the '50s with the work of [2], [3]. In the fixed confidence setting, the last decade has seen a flurry of activity providing new upper and lower bounds. In 2002, the *successive elimination* procedure of [4] was shown to find the best arm with order $\sum_{i \neq i_*} \Delta_i^{-2} \log(n\Delta_i^{-2})$ samples, where $\Delta_i = \mu_{i_*} - \mu_i$, coming within a logarithmic factor of the lower bound of $\sum_{i \neq i_*} \Delta_i^{-2}$, shown in 2004 in [5]. In 2012, algorithms referred to as *lower upper confidence bound (LUCB)* algorithms, motivated by the success of the upper confidence bound (UCB) style procedures of [6], were proposed that yielded sample complexities on the order of $\sum_{i \neq i_*} \Delta_i^{-2} \log(\sum_{j \neq i_*} \Delta_j^{-2})$ when applied to the best arm problem [1], [7]. In 2013, [8] proposed a procedure called *PRISM* which succeeds with $\sum_{i \neq i_*} \Delta_i^{-2} \log\log\left(\sum_{j \neq i_*} \Delta_j^{-2}\right)$ or $\sum_{i \neq i_*} \Delta_i^{-2} \log\left(\Delta_i^{-2}\right)$ samples depending on the parameterization of the algorithm, improving the result of [4] by at least a factor of $\log(n)$. In the same year, a procedure similar to *PRISM* called *exponential-gap elimination* was proposed by [9] which identifies the best arm using order $\sum_{i \neq i_*} \Delta_i^{-2} \log\log \Delta_i^{-2}$ samples, establishing the best known sample complexity result to date, coming within a doubly logarithmic factor of the lower bound of [5]. In late 2013, [10] pointed out that the sample complexity result of [9] was indeed optimal for at least the two armed case using a classical result of Farrell [11]. The Farrell result relies on the Law of Iterated Logarithm (LIL) which roughly states that $\limsup_t |\sum_{i=1}^t Z_i| / \sqrt{2t \log\log t} = 1$ almost surely where Z_i are standard normals. Inspired by the LIL, [10] proposed a UCB style algorithm using confidence bounds based on a finite version of the LIL that achieves the same optimal query complexity as the *exponential-gap elimination* procedure of [9].

Now that the community has two algorithms that obtain the optimal sample complexity for the best-arm problem in the fixed confidence setting, we feel that the time is right to summarize general principles of best-arm algorithms and present an empirical evaluation of those algorithms. This is of particular interest due to the fact that all of the algorithms discussed above use just one of three general sampling strategies: *action elimination*, *UCB*, or *LUCB* (while we consider *LUCB* to be a simple variant of *UCB*, the sampling strategy is just different enough to justify the distinction). The aim of this paper is to present a qualitative and quantitative overview of algorithms that use these sampling procedures. In addition, using techniques developed in [10], we present novel, concise proofs of near-optimal sample complexity results for the *action elimination*, *UCB*, and *LUCB* sampling strategies. The sample complexity result derived for *LUCB* is the best known result for the LUCB sampling strategy improving the result of $\sum_{i\neq i_*}\Delta_i^{-2}\log(\sum_{j\neq i_*}\Delta_j^{-2})$ [7] to $\sum_{i\neq i_*}\Delta_i^{-2}\log\left(n\log(\Delta_i^{-2})\right)$.

II. General Principles of Best-Arm Identification

Despite the multitude of algorithms and sample complexity results, all of the most popular algorithms can be described by essentially one of two algorithms. Similar observations have been made in the past, for instance in [12], but here we wish to succinctly describe the relationships between the algorithms' sampling strategies, provide intuition about how they work, and give sketches of their proofs using the LIL.

notation

We first define some notation. Without loss of generality, let the n arms be ordered such that $\mu_1 > \mu_2 \geq \cdots \geq \mu_n$ where $\mu_{i_*} = \mu_1$ and each $\mu_i \in [0,1]$. Also, define $\Delta_i = \mu_1 - \mu_i$. For any algorithm, let $X_{i,s}$, $s = 1, 2, \ldots$ denote independent samples from arm i where $\mathbb{E}[X_{i,s}] = \mu_i$ and $(X_{i,s} - \mu_i)$ is sub-Gaussian distributed. Let $T_i(t)$ denote the number of times arm i has been sampled up to time t and define $\widehat{\mu}_{i,T_i(t)} := \frac{1}{T_i(t)}\sum_{s=1}^{T_i(t)} X_{i,s}$ to be the empirical mean of the $T_i(t)$ samples from arm i up to time t. For any time t define

$$h_t = \arg\max_{i\in[n]} \hat{\mu}_{i,T_i(t)} \ , \quad \ell_t = \arg\max_{i\in[n]\setminus h_t} \hat{\mu}_{i,T_i(t)} + C_{i,t}$$

where $C_{i,t} > 0$ is typically derived from a tail-bound (e.g. Hoeffding's inequality) that may depend on $t, T_i(t), n$ and some confidence parameter δ.

The sampling strategies and their termination criteria are described as follows:

- **Action Elimination (AE) algorithm** - [2]–[4], [8], [9] Maintaining a set Ω_k for $k = 1, 2, \ldots$ initialized as $\Omega_1 = [n]$, these algorithms proceed in epochs by sampling the arms indexed by Ω_k a predetermined number of times r_k, and maintains arms according to the rule:

  # of epochs

  $$\Omega_{k+1} = \{i \in \Omega_k : \hat{\mu}_{a,T_a(t)} - C_{a,T_a(t)} < \hat{\mu}_{i,T_i(t)} + C_{i,T_i(t)}\}$$

  where $a \in \Omega_k$ is a reference arm (for instance $a = \arg\max_{i\in[n]} \hat{\mu}_{i,T_i(t)} + C_{i,T_i(t)}$). The algorithm terminates when $|\Omega_k| = 1$ and outputs the single element of Ω_k.

  In any algorithm, every arm must be sufficiently sampled before it can be decided with high probability that it is the best arm or not. This algorithm simply keeps sampling all the arms and throws those arms out that it is confident are not the best arm.

- **Upper Confidence Bound (UCB) algorithm** - [10], [13] Sample all arms once. For each each time $t > n$ the algorithm samples the arm indexed by

  $$\arg\max_{i\in[n]} \hat{\mu}_{i,T_i(t)} + C_{i,t}.$$

  One stopping condition is to stop when

  stopping condition

  $$\hat{\mu}_{h_t,T_{h_t}(t)} - C_{h_t,T_{h_t}(t)} > \hat{\mu}_{\ell_t,T_{\ell_t}(t)} + C_{\ell_t,T_{\ell_t}(t)} \qquad (1)$$

  and output h_t. Alternatively, one can stop when

  $$\exists i \in [n] \ : \ T_i(t) > \alpha \sum_{j\neq i} T_j(t) \qquad (2)$$

  and output $\arg\max_i T_i(t)$ for some $\alpha > 0$.

  While UCB sampling strategies were originally designed for the regret setting to optimize "exploration versus exploitation" [6], it was shown in [13] that UCB strategies were also effective in the pure exploration (find the best) setting. These algorithms are attractive because they are more sequential than the AE algorithms that tend to act more like uniform sampling for the first several epochs.

- **LUCB (a variation on UCB)** - [1], [7], [12] Sample all arms once. For each time $t > n$ sample the arms indexed by h_t and ℓ_t (i.e. at each time t two arms are sampled) and stop when the criterion defined in (1) is met.

  LUCB samples 2 a time h_t & ℓ_t

  While the LUCB and UCB sampling strategies appear to be only subtly different, we are motivated to discuss the LUCB strategies because they seem better designed for exploration than UCB sampling strategies. For instance, given just two arms, the most reasonable strategy would be to samepl both arms the same number of times until a winner could be confidently proclaimed, which is what LUCB would do. On the other hand, UCB strategies would tend to sample the best arm far more than the second-best arm leading to a strategy that seems to emphasize exploitation over pure exploitation. Our experiments demonstrate this effect.

In the remaining sections we provide a simple analysis of an instance of each of these algorithms and then present simulation results to show how they contrast in practice.

III. Analyses of Best Arm Algorithms

Here we define specifc instances of the general algorithms described in the previous section and give a simple proof sketch of their sample complexities. While the specific instances are known to not achieve the optimal sample complexity, they are all optimal up to $\log(n)$ factors and perform very well in practice. These specific instances were chosen to highlight the core components of the proofs and not bog the reader down in technicalities. The results rely on the finite form of the law of iterated logarithm of [10] and is restated here.

Lemma 1: Let $X_1, X_2, \ldots$ be i.i.d. sub-Gaussian random variables with scale[1] parameter $\sigma \leq 1/2$ and mean $\mu_i \in \mathbb{R}$. For any $\varepsilon \in (0,1)$ and $\delta \in (0, \log(1+\varepsilon)/e)$ one has with probability[2] at least $1 - \frac{2+\varepsilon}{\varepsilon/2}\left(\frac{\delta}{\log(1+\varepsilon)}\right)^{1+\varepsilon}$ that $\left|\frac{1}{t}\sum_{s=1}^{t} X_s - \mu_i\right| \leq U(t,\delta)$ for all $t \geq 1$ where $U(t,\delta) := (1+\sqrt{\varepsilon})\sqrt{\frac{(1+\varepsilon)t\log\left(\frac{\log((1+\varepsilon)t)}{\delta}\right)}{2t}}$.
While the algorithms and their analyses are inspired by previous works, the novelty here is the conciseness of the proofs

[1] The scale parameter was chosen such that analyses of algorithms that assumed realizations were in $[0,1]$ were still valid.

[2] The range on δ is restricted to guarantee that $\log(\frac{\log((1+\varepsilon)t)}{\delta})$ is well defined. This makes the analysis cleaner but in practice one can allow the full range of δ by using $\log(\frac{\log((1+\varepsilon)t+2)}{\delta})$ instead and obtain the same theoretical guarantees. This is done in our experiments.

finds best arm.
bounds exploration.

and the use of the LIL bound. For each algorithm we will show two events hold with high probability: 1) the algorithm terminates with no other arm than the best arm, and 2) the algorithm terminates after sampling all the arms no more than some constant depending on the problem parameters.

A. Action Elimination Algorithm

Consider the AE algorithm above with $r_k = 1$ so that at the end of the kth epoch we have that $T_i(t) = k$ for all $i \in \Omega_k$. Let $C_{i,k} := 2U(k, \delta/n)$ for all $i \in [n]$, and $a = \arg\max_{i\in\Omega_k} \widehat{\mu}_{i,T_i(t)}$. Applying Lemma 1 and a union bound we have

LIL

$$|\widehat{\mu}_{i,T_i(t)} - \mu_i| \leq U(T_i(t), \delta/n) \quad \forall i \in [n], \quad \forall t \geq 1 \tag{3}$$

with probability at least $1 - \frac{2+\varepsilon}{\varepsilon/2}\left(\frac{1}{\log(1+\varepsilon)}\right)^{1+\varepsilon}\delta$.

1) Algorithm terminates with the best arm: Conditioning on (3), if $i_* \in \Omega_k$ then with $a = \arg\max_{i\in\Omega_k} \widehat{\mu}_{i,T_i(t)}$,

$$\begin{aligned}\widehat{\mu}_{a,k} - \widehat{\mu}_{i_*,k} &= \widehat{\mu}_{a,k} - \mu_a + \mu_{i_*} - \widehat{\mu}_{i_*,k} - \Delta_a \\ &\leq 2U(k, \delta/n) = 2C_{a,k} = 2C_{i_*,k}\end{aligned}$$

which implies $i_* \in \Omega_{k+1}$. By induction we have that $i_* \in \Omega_k$ $\forall k \geq 1$ implying that if the algorithm terminates, it outputs the best arm.

2) Bounding total number of measurements: To bound the total number of samples, note that the ith arm is thrown out at the kth epoch if $\widehat{\mu}_{a,k} - \widehat{\mu}_{i,k} \geq 2U(k, \delta/n)$. By the definition of a and conditioning on (3) we have

$$\widehat{\mu}_{a,k} - \widehat{\mu}_{i,k} \geq \widehat{\mu}_{i_*,k} - \widehat{\mu}_{i,k} \geq -2U(k, \delta/n) + \Delta_i.$$

A straightforward calculation (see [10]) shows that

$$\begin{aligned}&\min\{k : U(k, \delta/n) \leq \Delta_i/4\} \\ &\qquad \leq \frac{2\gamma}{\Delta_i^2} \log\left(\frac{2\log(\gamma(1+\varepsilon)\Delta_i^{-2})}{\delta/n}\right)\end{aligned} \tag{4}$$

where $\gamma = 8(1+\sqrt{\varepsilon})^2(1+\varepsilon)$. We conclude that for any $\varepsilon \in (0,1)$ and $\delta \in (0, \log(1+\varepsilon)/e)$ we have with probability at least $1 - \frac{2+\varepsilon}{\varepsilon/2}\left(\frac{1}{\log(1+\varepsilon)}\right)^{1+\varepsilon}\delta$ that the described AE algorithm terminates with the best arm and has a sample complexity of order $\sum_{i\neq i_*} \Delta_i^{-2} \log\left(\frac{n\log(\Delta_i^{-2})}{\delta}\right)$, coming within a $\log(n)$ factor of optimum.

Using a tighter analysis, one can actually show that what prevents the removal of the suboptimal $\log(n)$ factor is the large deviations of $|\widehat{\mu}_{a,T_a(t)} - \mu_a|$ with $a = \arg\max_{i\in\Omega_k} \widehat{\mu}_{i,T_i(t)}$. Indeed, *PRISM* [8] and *exponential-gap elimination* [9] use a subroutine called *median elimination* developed in [4] to determine an alternative reference arm a which has smaller deviations and allows for the removal of the $\log(n)$ term. However, the analyses of those algorithms are more involved and beyond the scope of this paper. Unfortunately, the overhead coming in the form of constants using median elimination is prohibitively large for practical use, as we will show later in the experimental section.

B. UCB Algorithm

This analysis is inspired by [10]. Consider the UCB algorithm defined above with $C_{i,t} = (1+\beta)U(T_i(t), \delta/n)$ for some $\beta > 0$ and using the stopping condition defined in (2) with

$$\alpha = \left(\frac{2+\beta}{\beta}\right)^2 \left(1 + \frac{\log\left(2\log\left(\left(\frac{2+\beta}{\beta}\right)^2 n/\delta\right)\right)}{\log(n/\delta)}\right).$$

Once again we will condition on (3) since it holds with probability at least $1 - \frac{2+\varepsilon}{\varepsilon/2}\left(\frac{1}{\log(1+\varepsilon)}\right)^{1+\varepsilon}\delta$.

1) Algorithm terminates with the best arm: Assuming (3) holds and arm $i \neq i_*$ is played at time t, we have by definition that

$$\begin{aligned}\mu_i + (2+\beta)U(T_i(t), \delta/n) &\geq \widehat{\mu}_{i,T_i(t)} + (1+\beta)U(T_i(t), \delta/n) \\ &\geq \widehat{\mu}_{i_*,T_{i_*}(t)} + (1+\beta)U(T_{i_*}(t), \delta/n) \\ &\geq \mu_{i_*} + \beta U(T_{i_*}(t), \delta/n)\end{aligned} \tag{5}$$

which implies $(2+\beta)U(T_i(t), \delta/n) \geq \beta U(T_{i_*}(t), \delta/n)$. After some manipulation of this expression, we find that $T_i(t) \leq \alpha T_{i_*}(t)$. Thus, assuming (3) holds, the algorithm will never terminate with a suboptimal arm.

2) Bounding total number of measurements: Returning to (5), the expression also implies that $(2+\beta)U(T_i(t), \delta/n) \geq \Delta_i$. Solving for $T_i(t)$, we find that

$$T_i(t) \leq 1 + \frac{2\gamma}{\Delta_i^2} \log\left(\frac{2\log(\gamma(1+\varepsilon)\Delta_i^{-2})}{\delta/n}\right)$$

where $\gamma = (2+\beta)^2(1+\sqrt{\varepsilon})^2(1+\varepsilon)/2$. Using the fact that $T_{i_*}(t) = t - \sum_{i\neq i_*} T_i(t)$ we observe that i_* will eventually meet the stopping criterion resulting in a sample complexity of order $\sum_{i\neq i_*} \Delta_i^{-2} \log\left(\frac{n\log(\Delta_i^{-2})}{\delta}\right)$, coming within a $\log(n)$ factor of optimum. The β that optimizes the bounds is found to be equal to $\beta \approx 1.66$ but smaller values of β tend to work better in practice (see experiments section).

We remark that one can remove the sub-optimal $\log(n)$ term by running the same algorithm with $C_{i,t} = (1+\beta)U(T_i(t), \delta)$ and performing a more careful analysis as is done in [10]. The only difference in the analyses is that here we use the trivial lower bound $\alpha \sum_{j\neq i_*} T_j(t) \geq \alpha T_{i_*}(t)$ to make sure an arm $i \neq i_*$ is not output from the algorithm whereas a more careful analysis uses all the terms of the sum.

C. LUCB Algorithm

This analysis is inspired by [7]. Consider the LUCB algorithm defined above with $C_{i,t} = U(T_i(t), \delta/n)$ and using the stopping condition defined in (1) with $C_{i,t} = U(T_i(t), \delta/n)$.

1) Algorithm terminates with the best arm: Assuming (3) holds we trivially have that the stopping condition is only met with the best arm with probability at least $1 - \delta$.

2) Bounding total number of measurements: Define $c = (\mu_1 + \mu_2)/2$. We say arm i_* is BAD if $\hat{\mu}_{i_*,T_{i_*}(t)} - U(T_{i_*}(t), \delta/n) < c$ and an arm $i \neq i_*$ is BAD if $\hat{\mu}_{i,T_i(t)} + U(T_i(t), \delta/n) > c$. We claim that for all time $t \geq 1$

$$(3) \cap \{\hat{\mu}_{h_t,T_{h_t}(t)} - U(T_{h_t}(t), \tfrac{\delta}{n}) < \hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U(T_{\ell_t}(t), \tfrac{\delta}{n})\} \implies \{h_t \text{ is } BAD\} \cup \{\ell_t \text{ is } BAD\}. \tag{6}$$

In words, if the empirical means are well-behaved with respect to their confidence bounds and the algorithm has not yet terminated, then either h_t or ℓ_t has not been sufficiently sampled relative to c. This can be shown by contradiction by considering all possible combinations of ℓ_t, h_t being assigned to i_* or an arbitrary $i \neq i_*$ [7, Lemma 2].

For all $i \neq i_*$ define τ_i to be the first integer such that $U(\tau_i, \delta/n) \leq \Delta_i/4\}$ and define $\tau_{i_*} = \tau_2$. Assuming (3) holds, then for any $i \neq i_*$ and $s \geq \tau_i$

$$\begin{aligned}\hat{\mu}_{i,s} + U(s, \delta/n) &\leq \mu_i + 2U(s, \delta/n) \\ &= c + 2U(s, \delta/n) + \frac{(\mu_i - \mu_{i_*}) + (\mu_i - \mu_2)}{2} \\ &\leq c + 2U(s, \delta/n) - \frac{\Delta_i}{2} \leq c\end{aligned}$$

which implies that $i \neq i_*$ is *not* BAD. An analogous argument can be made for $i = i_*$ and τ_{i_*}.

Assuming (3) holds, by the above arguments we observe that the total number of rounds does not exceed

$$\begin{aligned}&\sum_{t=1}^{\infty} \mathbf{1}\{h_t \text{ is } BAD \text{ or } \ell_t \text{ is } BAD\} \\ &\quad= \sum_{t=1}^{\infty}\sum_{i=1}^{n} \mathbf{1}\left\{\{h_t = i \text{ or } \ell_t = i\} \cap \{i \text{ is } BAD\}\right\} \\ &\quad\leq \sum_{t=1}^{\infty}\sum_{i=1}^{n} \mathbf{1}\left\{\{h_t = i \text{ or } \ell_t = i\} \cap \{T_i(t) \leq \tau_i\}\right\} \leq \sum_{i=1}^{n} \tau_i\end{aligned}$$

where the last inequality holds by the fact that if $\{h_t = i \text{ or } \ell_t = i\}$ then $T_i(t+1) = T_i(t) + 1$ and this can only occur τ_i times before $T_i(t) > \tau_i$.

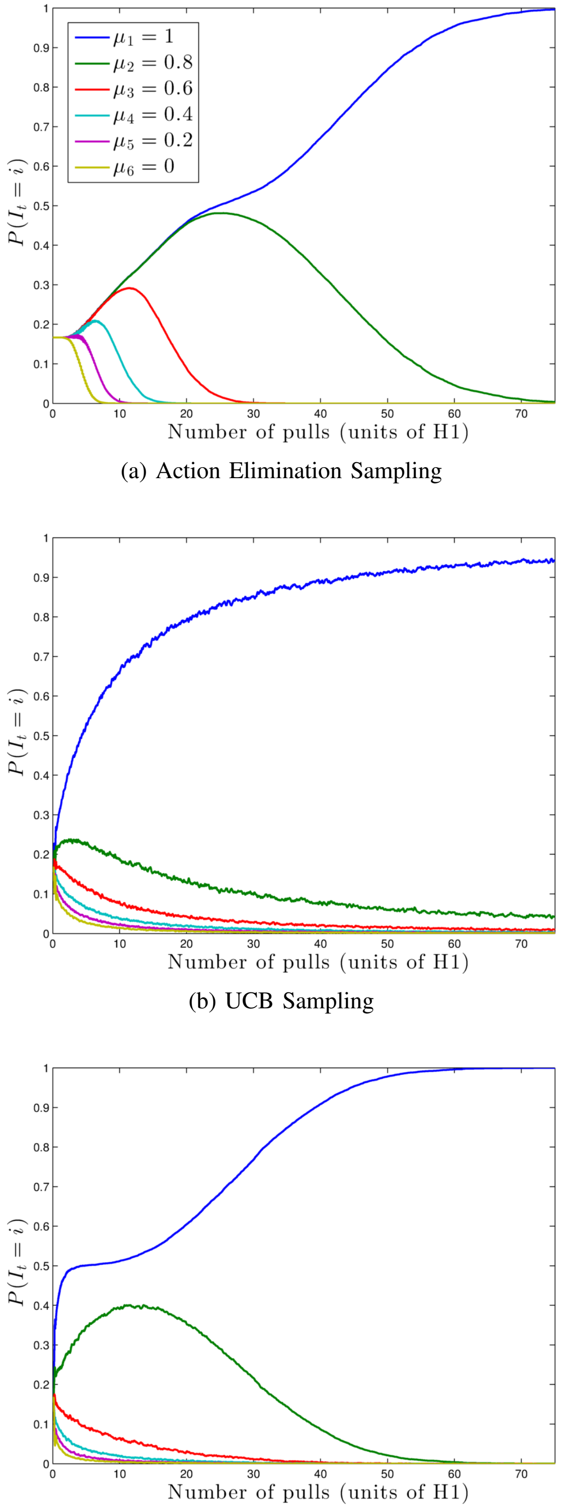
Plugging in the right-hand-side of (4) for τ_i and recalling that two samples are taken at each round we observe that with probability at least $1 - \frac{2+\varepsilon}{\varepsilon/2}\left(\frac{1}{\log(1+\varepsilon)}\right)^{1+\varepsilon}\delta$ the algorithm obtains a sample complexity of order $\sum_{i \neq i_*} \Delta_i^{-2} \log\left(\frac{n \log(\Delta_i^{-2})}{\delta}\right)$, coming within a $\log(n)$ factor of optimum.

We remark that the algorithm and proof of [7] that motivated the above analysis was originally derived for identifying the top m-arms and also yielded a non-trivial bound on the expected number of measurements, which our analysis does not. Using this analysis approach it is not clear how to remove the the $\log(n)$ factor and this is an interesting avenue of future work.

IV. Empirical Performance

Here we perform two sets of experiments. The first set of experiments contrasts the qualitative behavior of the *action elimination*, *UCB*, and *LUCB* sampling strategies. The second set of experiments is more quantitative and compares the stopping times of the state-of-the-art algorithms.

h = mean payoff.



(a) Action Elimination Sampling

(b) UCB Sampling

(c) LUCB Sampling

Fig. 1: Comparison of the three sampling strategies sampling arms $i = 1, 2, \ldots, 6$ with means μ_i. The algorithms used in the comparison were implemented as described in Section III.

Fig 1 results & info

A. Sampling Strategies: Action Elimination, UCB, LUCB

The above analyses of the *action elimination*, *UCB*, and *LUCB* strategies show that the sample complexities of the algorithms are very similar, even up to constants. And if we consider small, fixed values of n so that the effect of the $\log(n)$ term in the bounds is negligible, the algorithms are order optimal. We now explore how the algorithms differ, if at all, in how they sample the arms to decide the best arm. The algorithms used in the comparison were implemented as described in Section III.

implementation details

We chose to look at the very simple case of just $n = 6$ arms with linearly decreasing means: $\{1, 4/5, 3/5, 2/5, 1/5, 0\}$. All experiments were run with input confidence $\delta = 0.1$. And wherever the finite LIL is used, we use $\varepsilon = 0.01$. All realizations of the arms were Gaussian random variables with mean μ_i and variance1/4. To ignore the artifacts resulting from the action elimination algorithm taking multiple samples simultaneously, we estimate $\mathbb{P}(I_t = i)$ at every time t by calculating a proportion of the indices pulled in the interval $[t-n+1, t]$ and average over 5000 trials each algorithm completed.

The comparison of sampling procedures is plotted in Figure 1. Axes are plotted in units of $\mathbf{H}_1 := \sum_{i \neq i_*} \Delta_i^{-2}$ which is a dominant term in the sample complexity of best arm identification problems. We immediately observe a dramatic difference between the three sampling procedures: the action elimination strategy peels one arm away at a time and the plot of $\mathbb{P}(I_t = i)$ gives little indication of the best arm until many pulls in. On the other hand, the plot of $\mathbb{P}(I_t = i)$ for the LUCB and UCB sampling strategies clearly identifies the best arm very quickly with a large separation between the first and second arm. We remark that these algorithms may vary in performance using different parameters but the qualitative shape of these curves remain the same.

B. Stopping Time Comparison

In this section we investigate how the state-of-the-art methods for solving the best arm problem behave in practice. Before describing each of the algorithms in the comparison, we describe an LIL-based stopping criterion that can be applied to any of the algorithms.

LIL Stopping (LS) : For any algorithm and $i \in [n]$, we can apply the stopping condition of (1) with $C_{i,t} = U(T_i(t), \delta/n)$. Since (3) holds with probability at least $1 - \frac{2+\varepsilon}{\varepsilon/2}\left(\frac{1}{\log(1+\varepsilon)}\right)^{1+\varepsilon}\delta$ any procedure that stops with this criterion outputs the best arm with at least this probability.

The LIL stopping condition is somewhat naive but often quite effective in practice for smaller size problems when $\log(n)$ is negligible. To implement the strategy for any fixed confidence algorithm, simply run the algorithm with $\delta/2$ in place of δ and assign the other $\delta/2$ confidence to the LIL stopping criterion. The algorithms compared were:

- *Nonadaptive + LS* : Draw a random permutation of $[n]$ and sample the arms in an order defined by cycling through the permutation until the LIL stopping criterion is met.
- *Exponential-Gap Elimination (+LS)* [9] : This is an action elimination procedure that chooses a reference arm using a subroutine called *median elimination* [4]. The algorithm terminates when there is only one arm that has not yet been discarded (or when the LIL stopping criterion is met). This algorithm achieves the theoretical optimal sample complexity.
- *Successive Elimination* [4] : This is an action elimination known as *Successive Elimination*. This procedure uses $C_{i,k} = \sqrt{\log(\pi^2/3\ nk^2/\delta)/k}$.
- *lil'Successive Elimination* : This is the action elimination algorithm of Section III.
- *lil'UCB (+LS)* [10] : This is a UCB procedure and is run with $\beta = 1$, $a = (2+\beta)^2/\beta^2 = 9$, and $\delta = \left(\frac{\nu\varepsilon}{5(2+\varepsilon)}\right)^{1/(1+\varepsilon)}$ for input confidence ν. The algorithm terminates according to the first of (1) and (2). This algorithm achieves the theoretical optimal sample complexity.
- *LUCB1* [7]: This is an LUCB procedure run with $C_{i,t} = \sqrt{\frac{\log\left(\frac{405.5nt^{1.1}}{\delta}\log(\frac{405.5nt^{1.1}}{\delta})\right)}{2T_i(t)}}$ as prescribed in [12].
- *lil'LUCB* : This is the LUCB algorithm of Section III.

We did not compare to *PRISM* of [8] because the algorithm and its empirical performance are very similar to *Exponential-Gap Elimination* so its inclusion in the comparison would provide very little added value.

Three problem scenarios were considered over a variety problem sizes (number of arms). The "1-sparse" scenario sets $\mu_1 = 1/4$ and $\mu_i = 0$ for all $i = 2, \ldots, n$ resulting in a hardness of $\mathbf{H}_1 = 4n$. The "$\alpha = 0.3$" and "$\alpha = 0.6$" scenarios consider $n+1$ arms with $\mu_0 = 1$ and $\mu_i = 1 - (i/n)^\alpha$ for all $i = 1, \ldots, n$ with respective hardnesses of $\mathbf{H}_1 \approx 3/2n$ and $\mathbf{H}_1 \approx 6n^{1.2}$. That is, the $\alpha = 0.3$ case should be about as hard as the sparse case with increasing problem size while the $\alpha = 0.6$ is considerably more challenging and grows super linearly with the problem size. See [8] for an in-depth study of the α parameterization.

The stopping times of each algorithms are compared in Figure 2. Each algorithm was run on each problem scenario and problem size 50 times. The first observation is that *Exponential-Gap Elimination (+LS)* appears to barely perform better than uniform sampling with the LIL stopping criterion. This shows that despite this algorithm being order optimal, the constants in *median elimination* are just too large to make this algorithm practically relevant. Comparing the individual variants of the successive elimination and LUCB algorithms (that is, the originally derived bounds versus LIL bounds) we see that the LIL bounds provide a substantial improvement in performance. While the *lil'UCB+LS* algorithm seems to perform the best for large sparse problems, the *lil'LUCB* algorithm is the clear winner overall. This shows that n, and consequently the difference of $\log(n)$ between *lil'UCB* and *lil'LUCB*, would have to be very large to justify using the seemingly more exploitative but theoretically optimal *lil'UCB*.

V. Conclusion

This paper presented an overview of best-arm algorithms and provided some insight into how they work and are analyzed. Using the LIL we gave simple proofs of the sample complexities of instances of the three major sampling strategies. The sample complexity derived for *LUCB* is the best result to date for the LUCB sampling strategy.

Another family of algorithms that has received a lot of attention recently in the multi-armed bandit literature is *Thompson Sampling* [14]. This Bayesian style algorithm is of interest due to the ease in which side information about the problem structure can be encoded into the prior distribution on the means. While most of this work has been concentrated in the regret or Bayesian-regret scenarios, recent work suggests that Thompson sampling may also be effective at identifying the best arm and is a promising avenue of future work [15].

Acknowledgment

We thank Matt Malloy and Sébestian Bubeck for the many collaborations that contributed to this work.

References

[1] V. Gabillon, M. Ghavamzadeh, A. Lazaric *et al.*, "Best arm identification: A unified approach to fixed budget and fixed confidence," 2012.

[2] E. Paulson, "A sequential procedure for selecting the population with the largest mean from k normal populations," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 174–180, 1964.

[3] R. E. Bechhofer, "A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs," *Biometrics*, vol. 14, no. 3, pp. 408–429, 1958.

[4] E. Even-Dar, S. Mannor, and Y. Mansour, "Pac bounds for multi-armed bandit and markov decision processes," in *Computational Learning Theory*. Springer, 2002, pp. 255–270.

[5] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *The Journal of Machine Learning Research*, vol. 5, pp. 623–648, 2004.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[7] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone, "Pac subset selection in stochastic multi-armed bandits," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 655–662.

[8] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "On finding the largest mean among many," *arXiv preprint arXiv:1306.3917*, 2013.

[9] Z. Karnin, T. Koren, and O. Somekh, "Almost optimal exploration in multi-armed bandits," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[10] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "lil'ucb: An optimal exploration algorithm for multi-armed bandits," *arXiv preprint arXiv:1312.7308*, 2013.

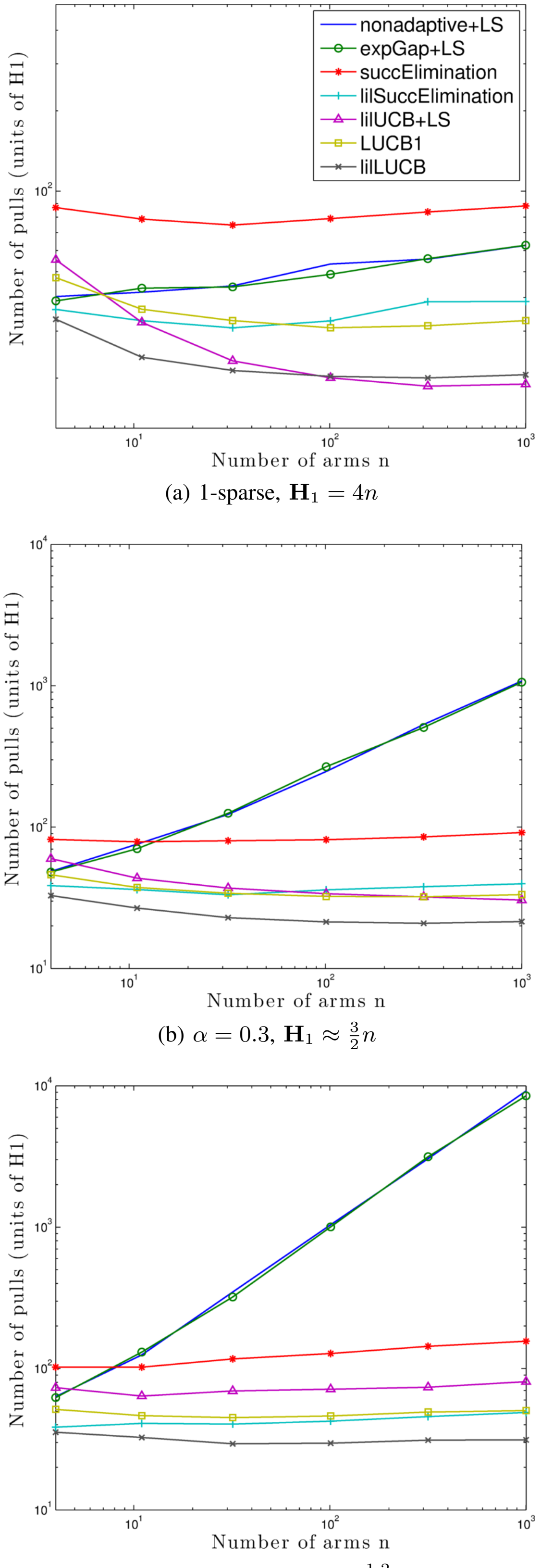
[11] R. H. Farrell, "Asymptotic behavior of expected sample size in certain one sided tests," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. pp. 36–72, 1964. [Online]. Available: http://www.jstor.org/stable/2238019

[12] E. Kaufmann and S. Kalyanakrishnan, "Information complexity in bandit subset selection." COLT, 2013.

[13] J.-Y. Audibert, S. Bubeck, and R. Munos, "Best arm identification in multi-armed bandits," *COLT 2010-Proceedings*, 2010.

[14] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," *arXiv preprint arXiv:1111.1797*, 2011.

[15] S. Bubeck and C.-Y. Liu, "Prior-free and prior-dependent regret bounds for thompson sampling," in *Advances in Neural Information Processing Systems*, 2013, pp. 638–646.



(a) 1-sparse, $\mathbf{H}_1 = 4n$

(b) $\alpha = 0.3$, $\mathbf{H}_1 \approx \frac{3}{2}n$

(c) $\alpha = 0.6$, $\mathbf{H}_1 \approx 6n^{1.2}$

Fig. 2: Stopping times of the algorithms for the three scenarios for a variety of problem sizes.

APPENDIX

Here we prove equation (6). Let $\mathcal{T}$ be the stopping time of the algorithm, i.e. the first time t such that $\{\hat{\mu}_{h_t,T_{h_t}(t)} - U(T_{h_t}(t),\delta/n) \geq \hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U(T_{\ell_t}(t),\delta/n)\}$. Then:

$$(3) \cap \{t < \mathcal{T}\} \implies \{h_t \text{ is } BAD\} \cup \{\ell_t \text{ is } BAD\}.$$

We will prove it by contradiction. Also, to reduce space, let $U[T_i(t)] = U(T_i(t),\delta/n)$. Assume h_t and ℓ_t are both not BAD and consider the disjoint events which all lead to contradictions:

Case 1:
$\{h_t = i_* \text{ is not } BAD\} \cap \{\ell_t \neq i_* \text{ is not } BAD\} \cap \{t < \mathcal{T}\}$

$$\begin{aligned}
&\implies \{\hat{\mu}_{h_t,T_{h_t}(t)} - U[T_{h_t}(t)] > c\} \cap \{\hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U[T_{\ell_t}(t)] < c\} \\
&\qquad \cap \{\hat{\mu}_{h_t,T_{h_t}(t)} - U[T_{h_t}(t)] < \hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U[T_{\ell_t}(t)]\} \\
&\implies \{\hat{\mu}_{h_t,T_{h_t}(t)} - U[T_{h_t}(t)] > \hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U[T_{\ell_t}(t)]\} \\
&\qquad \cap \{\hat{\mu}_{h_t,T_{h_t}(t)} - U[T_{h_t}(t)] < \hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U[T_{\ell_t}(t)]\} \\
&\implies \text{contradiction.}
\end{aligned}$$

Case 2:
$\{h_t \neq i_* \text{ is not } BAD\} \cap \{\ell_t = i_* \text{ is not } BAD\} \cap \{t < \mathcal{T}\}$

$$\begin{aligned}
&\implies \{\hat{\mu}_{h_t,T_{h_t}(t)} + U[T_{h_t}(t)] < c\} \cap \{\hat{\mu}_{\ell_t,T_{\ell_t}(t)} - U[T_{\ell_t}(t)] > c\} \\
&\implies \{\hat{\mu}_{h_t,T_{h_t}(t)} < \hat{\mu}_{\ell_t,T_{\ell_t}(t)}\} \\
&\implies \text{contradiction since } h_t = \arg\max_{i\in[n]} \hat{\mu}_{i,T_i(t)}\ .
\end{aligned}$$

Case 3:
$\{h_t \neq i_* \text{ is not } BAD\} \cap \{\ell_t \neq i_* \text{ is not } BAD\} \cap \{t < \mathcal{T}\}$

$$\begin{aligned}
&\implies \{\hat{\mu}_{h_t,T_{h_t}(t)} + U[T_{h_t}(t)] < c\} \cap \{\hat{\mu}_{\ell_t,T_{\ell_t}(t)} + U[T_{\ell_t}(t)] < c\} \\
&\implies \{\hat{\mu}_{i_*,T_{i_*}(t)} + U[T_{i_*}(t)] < c\} \\
&\implies \text{contradiction since (3) says } \hat{\mu}_{i_*,T_{i_*}(t)} + U[T_{i_*}(t)] \geq \mu_{i_*} \text{ and } \mu_{i_*} > c = \frac{\mu_1+\mu_2}{2}\ .
\end{aligned}$$